Simultaneous Localization and Mapping Based on (μ+1)-Evolution Strategy for Mobile Robots

Yuichiro Toda(✉) and Naoyuki Kubota

Graduate School of System Design, Tokyo Metropolitan University,
6-6 Asahigaoka, Hino, Tokyo, Japan
toda-yuuichirou@ed.tmu.ac.jp, kubota@tmu.ac.jp

Abstract. Simultaneous Localization and Mapping (SLAM) is one of the most important capabilities for autonomous mobile robots, and many researches have been proposed demonstrating the effective SLAM methods. However, these SLAM methods sometimes require assumptions such as the sensor model, which is difficult to implement and use the SLAM methods. In our previous work, a SLAM method based on Evolution Strategy (ES) was proposed and the on-line SLAM in indoor environments was realized. However, the definition of the map building method was not clear. Therefore, we propose a SLAM method based on a simple map building and search method. In this paper, we explain our autonomous mobile robot system and propose our SLAM method based on (μ+1)-ES. The experimental results show the effectiveness of the proposed method.

Keywords: SLAM · Occupancy grid map · Evolution strategy · Intelligent robotics

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental problem for an autonomous mobile robot because the robot searches an unknown environment and perform a decision making according to a situation in the environment [1]. Various types of methods for SLAM have been proposed such as Extended Kalman Filter (EKF) SLAM, Graph SLAM, visual SLAM. The EKF SLAM algorithm applies the EKF to online SLAM using maximum likelihood data associations. In the EKF SLAM, feature-based maps are used with point landmarks [2]. Graph SLAM solves a full SLAM problem in offline using all data obtained until the current time, e.g., all poses and all features in the map. Therefore, Graph SLAM has access to the full data when building the map [3,4]. Furthermore, cooperative SLAM (C-SLAM) has been also discussed in the study of multi-robot systems [5, 6].

In our previous work, we proposed a SLAM and initial self-localization method for multi-robot system based on the map sharing approach [7]. In the proposed method, one leader robot performs SLAM based on occupancy grid mapping. Both of localization and map building are performed by (μ+1)-Evolution Strategy (ES) [8]. However, our map building method is not stable because the map value uses an empty, occupied, partially occupied and unknown, and the definition of the partially occupied is not clear. Therefore, we apply a simple occupancy grid map method to our SLAM method for defining the map building method more clearly. Next, we propose the localization method based on (μ+1)-ES whose fitness function is composed of the summation of the map value and one penalty function for realizing on-line SLAM. In our SLAM, it is easy to implement the method, and our SLAM method can build the map and localize the robot position accurately.

This paper is organized as follows. Section 2 explains our robot system for an autonomous mobile robot. Section 3 proposes an SLAM method based on (μ+1)-ES. Finally, we show several experimental results of our SLAM method by using SLAM benchmark datasets.

2 SLAM

2.1 Mobile Robots

At first, we explain the hardware specification of a mobile robot (Table 1). We use an omni-directional robot with four omni-wheels and DC motors (Fig. 1). The robot can move to different omni-direction by changing the combination of output levels to motors. Basically, the action outputs of the robot are direct forward movement and rotation at the same position to avoid the slip appeared as noise in SLAM. Furthermore, the robot changes the moving direction only when the robot conducts obstacle avoidance. In addition, we use a laser range finder (LRF, URG04-LN) for SLAM.

Table 1. Specification of Omni-directional mobile robot

Diameter	300 mm
Height	177 mm
Weight	8 kg (approximately)
Maximal Speed	1.5 km/h
Operating Time (Battery)	1 hour
Maximal Payload Weight	15 kg
Communication Method	Wi-Fi (2.4 GHz)

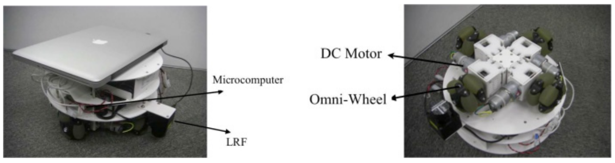

Fig. 1. Omni-directinal mobile robot

2.2 Procedure for SLAM

In our previous work, we use the occupancy grid map based on the following map value [9].

$$map(x,y)=\begin{cases}1 & \text{(occupied)}\\ 0.5 & \text{(partially occupied)}\\ 0 & \text{(unknown)}\\ -1 & \text{(empty)}\end{cases}. \tag{1}$$

Here the value of all cells is initialized at 0. Figure 2 shows the concept of the occupancy grid map. However, almost all cells excluding the empty and unknown cells are partially occupied cells because the cell including an object is defined as the partially occupied cell unless the object size fits the cell size correctly. Therefore, we use the simple definition of the occupancy grid map as follows;

$$map_t(x,y)=\frac{\mathrm{hit}_t(x,y)}{\mathrm{hit}_t(x,y)+\mathrm{err}_t(x,y)}. \tag{2}$$

where $\mathrm{hit}_t(x,y)$ and $\mathrm{err}_t(x,y)$ are the number of measurement and through points of LRF until the tth step, respectively. The measurement data is represented by (d_i, θ_i), i=1,2, ..., M, j=1,2, ..., L, where d_i is measurement distance from LRF; θ_i is the angle of the measurement direction; M is the number of total measurement directions; L_i ($=\left[\alpha^{Res}\cdot d_i\right]$) is the number of resolution for the map building by the occupancy grid model. Therefore, the map is updated by following procedure

Algorithm 1. Map-update

```
for $i$=1 to $M$ do
    for $j$=1 to $L_i$ do
        $u_{i,j}=\frac{j}{L_i}\left(d_i\cos(\theta_i+r_p)\right)+x_p$        (3)
        $v_{i,j}=\frac{j}{L_i}\left(d_i\sin(\theta_i+r_p)\right)+y_p$        (4)
        $x_{i,j}=\left[\alpha^{Map}\cdot u_{i,j}\right]$        (5)
        $y_{i,j}=\left[\alpha^{Map}\cdot v_{i,j}\right]$        (6)
        if $j = L_i$ then
            $\mathrm{hit}_t(x_{i,j}, y_{i,j}) = \mathrm{hit}_{(t-1)}(x_{i,j}, y_{i,j}) + 1$        (7)
        else
            $\mathrm{err}_t(x_{i,j}, y_{i,j}) = \mathrm{err}_{(t-1)}(x_{i,j}, y_{i,j}) + 1$        (8)
        endif
    endfor
endfor
```

where (x_p,y_p) is the position of the mobile robot; r_p is the posture; d_i is measurement distance from LRF in the ith direction; θ_i is the angle of the measurement direction; α^{MAP} is the scale factor mapping from the real world to the grid map.

Next, we explain out localization method. Basic localization algorithm is the almost same as out previous work. We apply (μ+λ)-ES for estimating the correct robot pose where μ and λ indicate the number of parent and offspring population generated in a single generation, respectively. ES can be used easily to discuss the formulation of strategy of artificial evolution in evolutionary computing [8]. We use (μ+1)-ES to enhance the local hill-climbing search as a continuous model of generations, which eliminates and generates one individual in a generation. A candidate solution is composed of numerical parameters of revised values to the current position ($g_{k,1}$, $g_{k,2}$) and rotation angle ($g_{k,3}$). In (μ+1)-ES, only an existing solution is replaced with the candidate solution generated by the crossover and mutation. We use the elitist crossover and adaptive mutation. Elitist crossover randomly selects one individual, and generates an individual by combining genetic information between the selected individual and best individual in order to obtain feasible solutions from the previous estimation result rapidly. Next, the following adaptive mutation is performed to the generated individual,

$$g_{k,h} \rightarrow g_{k,h} + \left(\alpha_h \cdot \frac{f_{\max} - f_k}{f_{\max} - f_{\min}} + \beta_h \right) \cdot N(0,1) \tag{9}$$

where f_k is the fitness value of the kth individual, f_{max} and f_{min} are the maximum and minimum of fitness values in the population; $N(0,1)$ indicates a normal random value; α_h and β_h are the coefficient and offset, respectively. A Fitness value of the kth candidate solution is calculated by the following equation,

$$\begin{aligned}
fit_k &= p_t^{occ}\left(x_{i,L}, y_{i,L}\right) \cdot \sum_{i=1}^{M} map_t\left(x_{i,L}, y_{i,L}\right) \\
p_t^{occ}\left(x_{i,L}, y_{i,L}\right) &= \frac{\sum_{i=1}^{M} \mathrm{hit}'_t\left(x_{i,L}, y_{i,L}\right)}{\sum_{i=1}^{M} \mathrm{hit}'_t\left(x_{i,L}, y_{i,L}\right) + \sum_{i=1}^{M} \mathrm{err}'_t\left(x_{i,L}, y_{i,L}\right)} \\
\mathrm{hit}'_t\left(x_{i,L}, y_{i,L}\right) &= \begin{cases} 1 & \text{if } \mathrm{hit}_t\left(x_{i,L}, y_{i,L}\right) > 0 \\ 0 & \text{else if } \mathrm{err}_t\left(x_{i,L}, y_{i,L}\right) > 0 \end{cases} \\
\mathrm{err}'_t\left(x_{i,L}, y_{i,L}\right) &= \begin{cases} 1 & \text{if } \mathrm{err}_t\left(x_{i,L}, y_{i,L}\right) > 0 \\ 0 & \text{else if } \mathrm{hit}_t\left(x_{i,L}, y_{i,L}\right) > 0 \end{cases}
\end{aligned} \tag{10}$$

where the summation of the map values is basic fitness value in (μ+1)-ES and p_t^{occ} indicates a penalty function. The summation of the map values is high if the estimation result is high. Furthermore, the penalty function has low value if many measurement points exist on empty cells. Therefore, this problem is defined as a maximization problem. Actually, we can estimate the robot pose by using only the summation of the map values. However, the estimation method sometimes gets stuck in local optima according to the environment if we use only the summation of the map value. Therefore, we use the penalty function p_t^{occ} for avoiding the situation. The localization based on (μ+1)-ES is finished when the number of iteration reaches the maximum number of iteration T. Algorithm 2 shows the total procedure of our SLAM method. Our SLAM procedure is very simple algorithm and it is easy to implement.

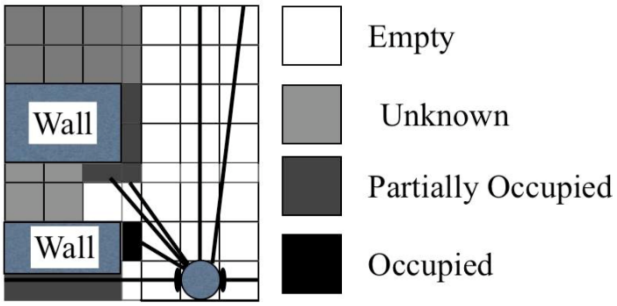

Fig. 2. Concept image of the occupancy grid map in our previous work [7]

```
Algorithm 2. Total Procedure of SLAM
1. $t = 0$, $\mathrm{hit}_t(x,y) = 0$ and $\mathrm{err}_t(x,y) = 0$
2. Input the LRF data
3. if $t \neq 0$ then
4.     Estimate the robot pose using ($\mu$+1)-ES
5. end if
6. Perform Map-update
7. $t = t + 1$
8. return to step 1
```

3 Experimental Results

We conducted an experiment of the proposed method by using two SLAM benchmark datasets [10,11]. We used only measurement data of LRF form these datasets. Figure 3 shows the ground truth of each dataset. In this experiment, we used two conditions. Condition 1 used the fitness function with p_t^{occ}. Condition 2 used the fitness function without p_t^{occ}. Our proposed algorithm was run on 3.5GHz 6-Core Intel Xeon E5 processor. Table xx shows the parameters using these experiments.

Figure 4 and 5 show an example of the experimental result of Condition 1 in each dataset. In these results, our proposed method can correctly localize and build the map compared to Fig. 4 and 5. Figure 6 shows the transition of variance of the best fitness value in each time step. The variance values are stable between until about the 1400th step in both results. On the other hand, Condition 1 is more stable than Condition 2 from about the 1400th step. Fig. 7 shows a failed example of Condition 2. In Fig. 7, red circles mean that the localization is often failed in these areas because (μ+1)-ES gets stuck in local optima in these areas if the fitness function does not include the penalty function p_t^{occ}. Table 3 shows experimental results of computational time in each dataset. In both results, the average of the computational time is about 18 [ms], and we consider that this computational time is enough for on-line SLAM. In this way, SLAM based on (μ+1)-ES can build the map and localize the robot position by designing the suitable fitness function according to the map building method.

Table 2. Setting parameters for the experiments

Number of trial in each experiment	10
Number of parents μ	100
Maximum number of iterations T	1000
Coefficients for adaptive mutation α_1, α_2	10.0
Coefficients for adaptive mutation α_3	1.0
Offset for adaptive mutation β	0.01
Cell size α^{Map}	100

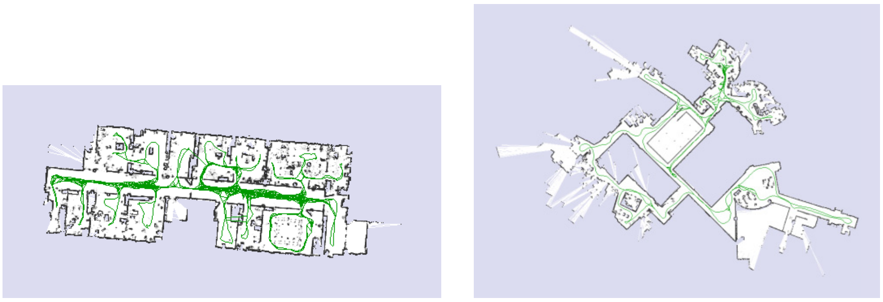
(a) Freiburg Indoor Building 079 (b) MIT CSAIL Building

Fig. 3. Correct map building result using the benchmark datasets [11].

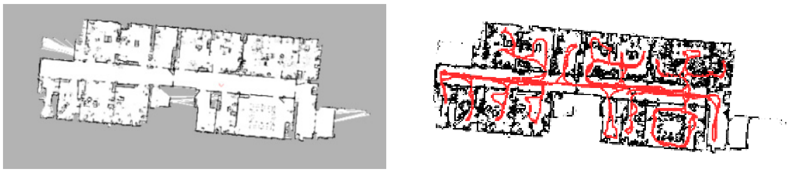
(a) Map building (b) Localization

Fig. 4. Experimental results of map building and localization in Freiburg Indoor Building 079. In (b), red line indicates localization result.

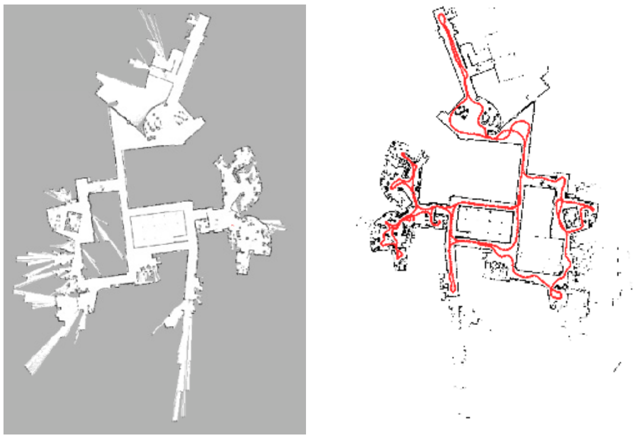

Fig. 5. Experimental results of map building and localization in MIT CSAIL Building. In (b), red line indicates localization result.

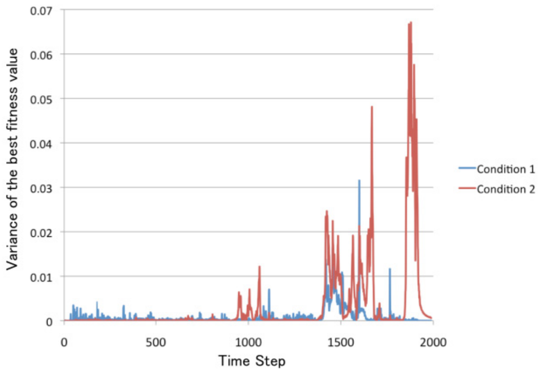

Fig. 6. Variance of the best fitness value in MIT CSAIL Building.

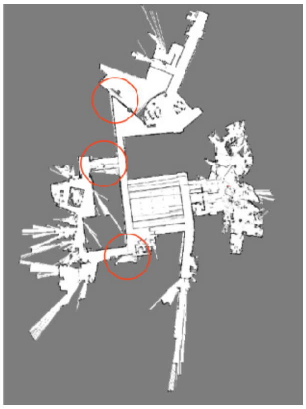

Fig. 7. A failed example of Condition2.

Table 3. Experimental results of computational time [ms]

	Freiburg Indoor Building 079	MIT CSAIL Building
Average	18.8	18.6
Variance	3.71	3.75
Max	28	26
Min	12	11

4 Summary

In this paper, we proposed our SLAM based on Evolution Strategy (ES). Our SLAM method changed the definition of the occupancy and grid map. Next, we proposed our localization method based on (μ+1)-ES. The fitness function of the ES used the summation of the map value and one penalty function based on occupied and empty cells. The experimental results showed that our SLAM method realized on-line SLAM by using the SLAM benchmark datasets. However, our SLAM method sometimes gets stuck in local optima because the search strategy of (μ+1)-ES is based on hill climbing search. Therefore, we will change the search method for combining the local and global search based on ES.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press (2005)
2. Huang, G.P., Trawny, N., Mourikis, A.I., Roumeliotis, S.I.: Observability-based consistent EKF estimators for multi-robot cooperative localization. Journal Autonomous Robots **30**(1), January 2011
3. Folkesson, J., Christensen, H.I.: Closing the Loop With Graphical SLAM. IEEE Transactions on Robotics **23**(4), 731–741 (2007)
4. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Incremental Smoothing and Mapping. IEEE Transactions on Robotics **24**(5), 1365–1378 (2008)
5. Pinheiro, P., Wainer, J.: Planning for multi-robot localization. In: da Rocha Costa, A.C., Vicari, R.M., Tonidandel, F. (eds.) SBIA 2010. LNCS, vol. 6404, pp. 183–192. Springer, Heidelberg (2010)
6. Choi, J., Choi, M., Nam, S.Y., Chung, W.K.: Autonomous topological modeling of a home environment and topological localization using a sonar grid map. Journal Autonomous Robots **30**(4), May 2011
7. Toda, Y., Kubota, N.: Self-localization Based on Multi-resolution Map for Remote Control of Multiple Mobile Robots. IEEE Transactions on Industrial Informatics **9**(3), 1772–1781 (2013)
8. Fogel, D.B.: Evolutionary Computation. IEEE Press (1995)
9. Thrun, S.: Learning Occupancy Grid Maps With Forward Sensor Models. Autonomous Robots, Springer **25**(2), 111–127 (2003)
10. Burgard, W., Stachniss, C., Grisetti, G., Steder, B., Kummerle, R., Dornhege, C., Ruhnke, M., Kleiner, A., Tardos, J.D.: A comparison of SLAM algorithms based on a graph of relations. In: Intelligent IEEE/RSJ International Conference on Robots and Systems, IROS 2009, pp. 2089–2095, October 10-15, 2009
11. slam benchmarking. http://kaspar.informatik.uni-freiburg.de/~slamEvaluation/index.php